Classification of Chemical compounds using Deep Learning using RNN

**Question:-** The given dataset contains details about organic chemical compounds including their chemical features, isomeric conformation, names and the classes in which they are classified. The compounds are classified as either ‘Musk’ or ‘Non-Musk’ compounds. Your task is to build a classification model on the given data using any Deep Learning approach that you deem appropriate viz. Multi-Layer Perceptron, CNN, RNN, etc. or you could also use transfer learning approaches through selection of appropriate pre-trained model. The data has to be split in an 80:20 ratio for training and validation datasets. You can perform whatever preprocessing and post-processing operations on the data that may help you improve the performance of your model. You re required to report the performance measures of the model viz. Accuracy( Training and Validation) and Loss(Training and Validation) graphs, F1 score, precision, recall, etc. along with a well detailed report of what models, pre-processing, post-processing approaches you have used and why you chose to use these approaches.

**Data Pre-Processing Steps**

After loading the data:

1. The data is checked for any duplicate rows.
2. Then the loaded data is check for presence of null values in the data. There are no null values in the dataset.
3. The “ID” column is removed from the dataset and also the “molecule\_name” is subjected to the Label Encoder function.
4. Once the missing values analysis is done, the data is then checked for outlier analysis. On plotting the box plot of all the columns, it was observed that following columns have the outlier data:- f1, f2, f5, f6, f7, f10, f12, f13, f14, f15, f16, f18, f19, f20, f21, f28, f29, f31, f33, f35, f36, f37, f42- f51, f54, f56, f58, f59, f61, f63, f66-f68, f70, f73 - f76, f78 - f80, f38, f87,f88, f92 - f95, f102 - f108, f110 – f112, f116, f124, f126, f127, f129, f131-f136, f138, f140, f141, f145- f149, f151, f153, f155, f157, f160- f166
5. The outlier data is removed with the help of Z-Score analysis.
6. Once the outlier data is removed, then the data is split into 3 different datasets
7. Training Data : X\_train , y\_train
8. Validation Data: X\_val, y\_val
9. Testing Data: X\_test, y\_test

**Model Training**

*model = keras.Sequential([*

*keras.layers.Flatten(input\_shape=(167,)),*

*keras.layers.Dense(8, activation=tf.nn.sigmoid),*

*keras.layers.Dense(4, activation=tf.nn.relu),*

*keras.layers.Dense(1, activation=tf.nn.sigmoid)*

The Sequential model is a linear stack of layers.

**Specifying the input shape**

The model needs to know what input shape it should expect. For this reason, the first layer in a Sequential model needs to receive information about its input shape. In this case there are abput 167 input features

First Activation Function used is sigmoid with 8 hidden layers.

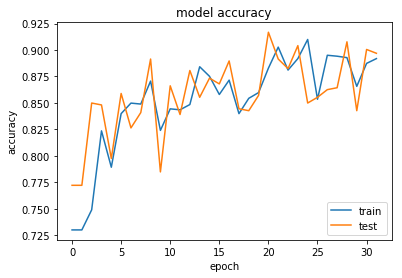
Second Activation Function used is relu (Rectified Linear Unit) with 4 hidden layers.

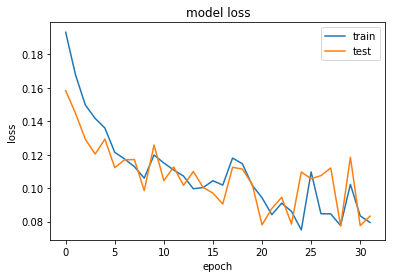
Third Activation Function used is sigmoid with 1 hidden layer.

**Compilation**

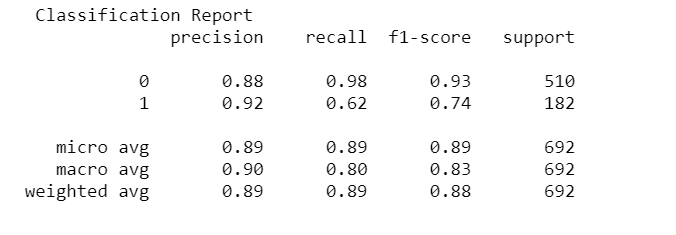
On compiling the model , optimizer used is adam, loss function used is mse (Mean Square Error) and metrics used is accuracy.

Model Loss and Accuracy graph are shown below





Classification Report:



Confusion Matrix:

